Offline Reinforcement Learning with Imputed Rewards

Anonymous authors
Paper under double-blind review

Abstract

Offline Reinforcement Learning (ORL) offers a robust solution to training agents in applications where interactions with the environment must be strictly limited due to cost, safety, or lack of accurate simulation environments. Despite its potential to facilitate deployment of artificial agents in the real world, Offline Reinforcement Learning typically requires very many demonstrations annotated with ground-truth rewards. Consequently, state-of-the-art ORL algorithms can be difficult or impossible to apply in data-scarce scenarios. In this paper we propose a simple but effective Reward Model that can estimate the reward signal from a very limited sample of environment transitions annotated with rewards. Once the reward signal is modeled, we use the Reward Model to impute rewards for a large sample of reward-free transitions, thus enabling the application of ORL techniques. We demonstrate the potential of our approach on several D4RL continuous locomotion tasks. Our results show that, using only 1% of reward-labeled transitions from the original datasets, our learned reward model is able to impute rewards for the remaining 99% of the transitions, from which performant agents can be learned using Offline Reinforcement Learning.

1 Introduction

Deep Reinforcement Learning (Wang et al., 2024) is notoriously sample inefficient because agents can require hundreds of millions of interactions with the environment for convergence. This is prohibitive in many – if not most – real-world applications where interactions with the environment are expensive, dangerous, impossible or all of the above. The prohibitive sample complexity of Deep Reinforcement Learning is a barrier to its application in many scenarios where high-fidelity environment simulators are unavailable.

One of the most promising recent trends in Deep Reinforcement Learning aimed at mitigating the sample inefficiency of traditional approaches is Offline Reinforcement Learning (Levine et al., 2020; Prudencio et al., 2023). Offline RL, sometimes also called Batch Reinforcement Learning, aims to learn policies from a fixed dataset of sampled agent-environment interactions, which turns the Reinforcement Learning problem into a more tractable *supervised* learning problem. Datasets for Offline Reinforcement Learning typically contain state-action-reward-next state tuples, collected either from human experts interacting with the environment or from random exploration. However, Offline Reinforcement Learning algorithms assume that the total reward function is known and well-defined. This assumption may not be valid in many real-world scenarios, where the reward function may be noisy, sparse, or even missing, thus limiting the applicability of Offline RL. To deal with the problem of missing reward, Imitation Learning (Zare et al., 2023; Gavenski et al., 2024) and Inverse Reinforcement Learning (Russell, 1998; Arora & Doshi, 2021) can be used as alternative approaches, but their application introduces new constraints that must be handled appropriately.

Imitation learning allows an agent to learn a policy by observing and imitating the behavior of an expert, rather than relying solely on environmental rewards. This approach is particularly useful when the reward signal is unreliable, poor, or even absent, as it allows the agent to infer the correct

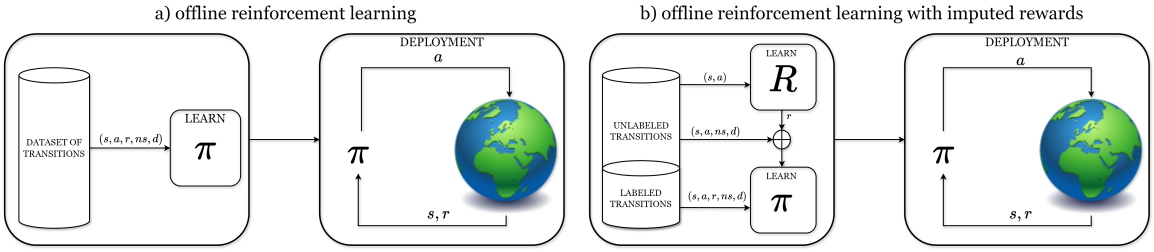

Figure 1: Illustration of the scenarios of interest. (a) Classical offline reinforcement learning solutions are trained with a large set of transitions in which the reward signal is fully defined. On the other hand, in real applications, the reward signal may be available for only a small fraction of the total transitions, as in (b). In this case, the ORL algorithms are forced to use only those transitions where the reward signal is present, because of their inability to exploit the entire distribution, unless the reward signal is modeled from the distribution of reward-labeled transitions.

actions simply by emulating the expert. The requirement for expert knowledge often cannot be met for realistic applications, limiting agents to less-than-optimal behavior. Inverse Reinforcement Learning , on the other hand, takes a different approach to learning from expert behavior. Instead of directly imitating actions, like in Imitation Learning, IRL aims to infer the underlying reward function that the expert is implicitly optimizing. Once this reward function is estimated, standard Reinforcement Learning techniques can be used to find an optimal policy for the learned reward. Despite the ability of Imitation and Inverse Reinforcement Learning techniques to alleviate the missing reward problem, they implicitly assume the presence of optimal demonstrations or unlimited access to the environment. Compared with existing solutions, the Reward Model we propose is free from all previous requirements: the reward signal is modeled from a mere distribution of reward-labeled transitions of unknown quality via supervised learning. Therefore, the environment is never accessed and no prior knowledge about the underlying distribution is required.

We focus on developing an effective technique to alleviate the limitations of Offline Reinforcement Learning in scenarios with highly unbalanced datasets – that is, in environments in which the reward signal is defined for only a small portion of the available transitions. Our reward model is a simple two-layer MLP which, when trained using only 1% of the available transitions, can be used to impute rewards for the remaining 99% of the dataset. This new dataset (with 99% of the transitions using imputed rewards) is then used to perform Offline RL. This approach resulted in a consistent increase in ORL performance in terms of average D4RL scores for TD3BC (Fujimoto & Gu, 2021) and IQL (Kostrikov et al., 2022) agents, compared to applying offline RL in data sparsity scenarios.

The rest of the paper is organized as follows. In the next section, we discuss recent work from the literature most related to our contribution. In Section 3 we introduce our reward modeling and reward imputation framework, and in Section 4 we report results of experiments performed on the D4RL MuJoCo continuous locomotion tasks. We conclude in Section 5 with a discussion of our contribution.

2 Related Work

Behavior Cloning (Pomerleau, 1988), is based on a supervised learning approach and is the simplest Imitation Learning algorithm. Due to its way of modeling a policy, its application is limited to datasets of optimal (or near optimal) experiences. Moreover, a Behavior Cloning agent performs best when presented with observations close to the training set distribution; otherwise, its performance collapses (Reddy et al., 2019). A solution to this problem was proposed by DAgger (Ross et al., 2010) in which the agent leverages its own induced state distribution to reduce the influence of out-of-distribution states. However, DAgger requires human intervention to relabel the input distribution. The high sample complexity and the requirement for expert demonstrations restrict the

wide application approaches based on Behavior Cloning. In contrast, our solution enables Offline Reinforcement Learning by modeling the reward signal without requiring either the intervention of human expert operators or expert-quality transitions.

An alternative is offered by Inverse Reinforcement Learning (Russell, 1998) approaches in which an IRL algorithm, from the expert distribution of demonstrations, can model the underlying reward signal implicitly optimized, which is essential for policy learning through interactions with the environment. IRL agents potentially offer more solid performance than BC mimics by engaging the learning process from a Reinforcement Learning point of view. However, the learning process outlined by IRL agents has its own limitations. With respect to BC approaches, IRL agents are eager for computational resources due to their interactions with the environment. Moreover, the typical trial-and-error learning process of Reinforcement Learning algorithms can be dangerous or unfeasible to conduct in certain real-world scenarios. The sample complexity problem of IRL agents has been addressed by implementing generative approaches as in (Ho & Ermon, 2016). But once again, the presence of optimal/expert demonstrations is required. In our solution, we model the reward signal in an offline setting, thus avoiding any interaction with the environment or particular requirements for optimal/expert demonstrations.

Offline Inverse Reinforcement Learning tries to alleviate the above limitations. Offline IRL approaches can be roughly categorized into two branches: adversarial approaches (Jarboui & Perchet, 2021; Kostrikov et al., 2019) that leverage GANs (Goodfellow et al., 2014) to minimize the distance between the agent and the expert behaviors but suffer from vanishing gradient and convergence problems (Arjovsky & Bottou, 2017), and reward learning approaches (Klein et al., 2013; Zolna et al., 2020) that, on the other hand, imply running ORL solutions in an inner loop, thus increasing the overall complexity. An alternative approach is offered by Xu & Zhan (2022) that, by contrasting expert with suboptimal demonstrations, add a weighting factor for the Behavioral Cloning term, but it requires extra effort to distinguish transitions based on their quality. The Reward Model we propose is able to impute rewards for reward-free transitions without the need to know the quality of the underlying distribution, thus addressing the modeling process only via label supervision.

3 Imputing Rewards for Offline Reinforcement Learning

3.1 Preliminaries

Reinforcement Learning (Sutton & Barto, 1998) aims to control a dynamic system, which is defined by a Markov Decision Process (MDP). The Markov Decision Process is defined by a tuple $\mathrm{M} = (S, A, T, d_0, r, \gamma)$: S is the state space that could be continuous or discrete; A is the action space that could also be continuous or discrete; T represents the transition function that maps a *(state, action)* pair at time t into the state at next timestep, $T(s_{t+1}|s_t, a_t)$; d_0 is the initial state distribution $d_0(s_0)$ that indicates in what state the environment is initialized; r is the reward function defined as $r : SxA \rightarrow \mathrm{R}$ that represents the feedback signal for taking action a in the state s; $\gamma\epsilon(0, 1]$ is the discount factor that weights short-term rewards with respect to long-term rewards.

The objective of Reinforcement Learning is to learn the parameters of an agent π to maximize the expected future discounted reward $J(\pi)$ obtained by the agent in environment M:

$$J(\pi) = \mathrm{E}_{\tau \sim p_\pi(\tau)} \left[\sum_{t=0}^{H} \gamma^t r(s_t, a_t) \right], \tag{1}$$

where γ is a discount factor controlling the trade off between immediate and future reward. Reinforcement learning approaches typically require millions of interactions with the environment in order to minimize this objective J, given that the environment dynamics T and reward signal r are typically unknown.

Offline Reinforcement Learning is a data-driven approach to the RL problem. In this setting, the goal is still to optimize the objective function J, but without access to the environment. Instead, the agent

is trained on static dataset of transitions $D = (s_t^i, a_t^i, s_{t+1}^i, r_t^i)_{i=1}^N$. An Offline RL algorithm should maximize J having access only to this restricted training set of labeled environment transitions. In this work we consider scenarios in which the *labeled* transitions available are very few, but we assume access to a large number of *unlabeled* transitions (i.e. no instantaneous reward is available) on which we impute rewards.

3.2 Methodology

We consider Offline RL scenarios in which only a very small sample of reward-labeled transitions are available. That is, the training dataset consists of a labeled subset of N_1 transitions labeled with rewards:

$$\mathcal{D}_{\updownarrow} = \{(\mathbf{s}_n, \mathbf{a}_n, r_n)\}_{n=1}^{N_l}, \tag{2}$$

and a much larger set of unlabeled transitions:

$$D_u = \{(\mathbf{s}_n, \mathbf{a}_n)\}_{n=1}^{N_u} \tag{3}$$

The Reward Model MLP $f(\mathbf{x}, \mathbf{a}; \theta)$ is trained using supervised learning on a dataset $\mathcal{D}_{\updownarrow}$ containing state-action-reward triples to minimize the expected Mean Squared Error on the training set $\mathcal{D}_{\updownarrow}$:

$$\mathcal{L}(\mathcal{D}_{\updownarrow}, \theta) = \mathbb{E}_{(\mathbf{x},\mathbf{a},r)\sim\mathbf{D_l}}\left[(f(\mathbf{x}, \mathbf{a}; \theta) - r)^2\right]. \tag{4}$$

We then use this trained Reward Model to impute missing rewards in D_u and construct the Offline RL dataset with imputed rewards:

$$D = D_l \cup \bigcup_{(\mathbf{x},\mathbf{a})\in D_u} \{(\mathbf{x}, \mathbf{a}, f(\mathbf{x}, \mathbf{a}; \theta)\} \tag{5}$$

This dataset with imputed rewards can be used with any offline RL algorithm to train an agent. In the next section, to address data scarcity scenarios, we evaluate agents in which D_l contains only 1% of reward-labeled transitions. Consequently, the final offline dataset D consists of 99% of rewards imputed via our Reward Model.

4 Experimental Results

In this section we report on a range of experiments we performed to evaluate the effectiveness of imputing rewards for Offline Reinforcement Learning using out reward model.

4.1 Environments and Datasets

We use D4RL (Fu et al., 2020) continuous locomotion environments based on MuJoCo (Todorov et al., 2012). D4RL provides offline RL datasets for these simulated environments, and have become standard benchmarks for Offline RL. We report results on the are Halfcheetah, Walker2D, and Hopper environments. For each environment we consider three variants of offline experience, each collected by a Reinforcement Learning agent at different points of its training::

- **Medium-Replay**: this dataset consists of all experience the agent collected during its training until achieving Medium quality. This implies that it also contains random or near-random trajectories, corresponding to the initial steps of the training process. This dataset is typically difficult for Behavior Cloning approaches.
- **Medium**: This dataset is a collection of trajectories collected by an RL agent trained to almost half of its maximum ability.

Environment	Baseline (100%)			1%		1% + Imputed	
	TD3BC	IQL	BC	TD3BC	IQL	TD3BC	IQL
halfcheetah-medium-v2	48.30	47.40	42.60	10.03	18.28	48.50	44.46
hopper-medium-v2	59.30	66.30	52.90	11.32	15.53	58.13	36.47
walker2d-medium-v2	83.70	78.30	75.30	6.00	1.21	82.81	69.87
halfcheetah-medium-replay-v2	44.60	44.20	36.60	2.60	13.38	44.71	32.73
hopper-medium-replay-v2	60.90	94.70	18.10	5.11	4.56	52.46	26.42
walker2d-medium-replay-v2	81.80	73.90	26.00	5.37	1.44	69.15	47.38
halfcheetah-medium-expert-v2	90.70	86.70	55.20	21.84	19.28	87.88	51.77
hopper-medium-expert-v2	98.00	91.50	52.50	11.62	21.84	83.65	22.94
walker2d-medium-expert-v2	110.10	109.60	107.50	8.77	8.04	89.30	107.38
Gym locomotion-v2 average	*75.27*	*76.96*	*51.77*	*9.19*	*11.51*	*68.51*	*48.82*

Table 1: The normalized results on D4RL Gym. *Baseline* results are taken from the IQL paper (Kostrikov et al., 2022). The results for the *1%* and *1% + Imputed* scenarios are calculated by averaging mean returns over 10 evaluation trajectories and five random seeds.

- **Medium-Expert**: This dataset is a mixture of trajectories from the Medium and the Expert datasets.

Non-expert quality distributions have mostly incomplete trajectories, with different degrees of exploration of the state-action space. As quality increases, highly rewarding regions are encountered more frequently.

4.2 Training Details

Our Reward Model is trained to minimize Equation 4 for 300 epochs, using the Adam optimizer with a learning rate of $1e^{-4}$. To make a fair comparison, we used the parameters of both the original implementations of IQL and TD3BC.

4.3 Baselines and Performance Evaluation

To quantify the performance of our solution, we compare the scores obtained by the IQL and TD3BC agents on different distributions. Each reported score represents the average of the normalized rewards, also called D4RL scores, collected during the evaluation in the last epoch of the offline training process. Each evaluation consists of ten episodes (1000 steps each) of online interactions with the environment. From each of these ten episodes, we take the normalized average of the last ten transitions. The whole process is repeated for five random seeds.

4.4 Comparative Performance Evaluation

In Table 1, we show the final performance obtained from agent training with the following settings:

- **Baseline**: the original datasets are used to measure the upper bounds;
- **1%**: each ORL agent is trained only on transitions where the reward signal is available and well defined. In our scenario, this represents only 1% of the transitions in the original dataset, previously called the train set;
- **1% + Imputed**: each ORL agent is trained on the augmented dataset: from the previous scenario (1%) the transitions where the reward signal was initially removed (99%), but then reconstructed via our Reward Model, are combined.

From the column *1%*, we can see that, on average, the IQL algorithm performs better than TD3BC in the *data scarcity* scenario. Nonetheless, for both algorithms, the scores obtained by feeding the agents with such small datasets are far from the baselines. For Medium and Medium Replay

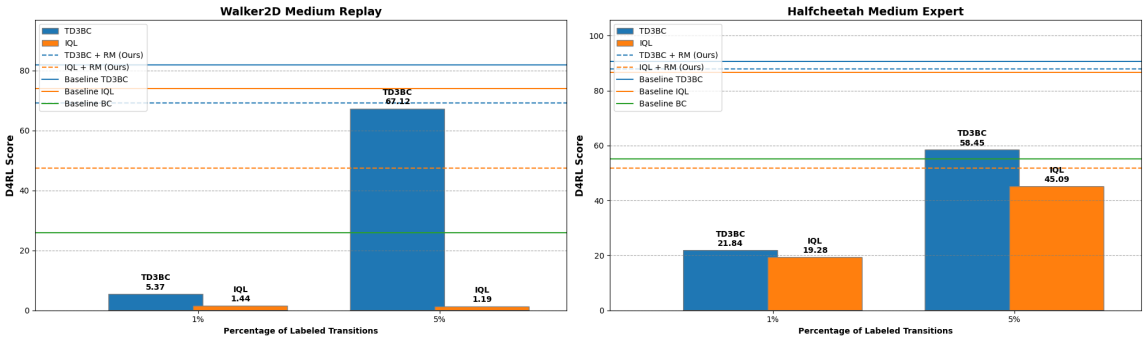

Figure 2: Visual comparison of TD3BC and IQL results by changing the input distribution from 1% to 5% of reward-labeled transitions, for the Walker2D Medium Replay and Halfcheetah Medium Expert scenarios. In blue, we have the scores achieved by the TD3BC algorithm, whereas in orange, the IQL scores. The solid lines indicate the baseline scores of each algorithm trained on the original dataset. The dashed lines show the performance of our solution: only 1% of reward-labeled transitions from the original dataset are considered, while the remaining 99% of transitions are labeled by imputing the reward signal via our Reward Model. The solid green lines represent the score achieved by running BC agents on the original dataset.

distributions in the Halfcheetah environment with one percent of total transitions, IQL's application is more robust than TD3BC, although in the baseline scenario, TD3BC scores are better in that environment. In the Hopper environment, TD3BC is able to achieve better scores only in the Medium Replay distribution, while, on the other contrary, in the Walker2D environment, IQL fails.

However, through the application of our Reward Model, the outcomes flip. Whenever in the 1% scenario the TD3BC fails, now by accessing a wider distribution, this algorithm offers stronger performance, even better than the IQL counterpart. Catastrophic failures happen for the TD3BC algorithm with the Walker2D Medium and Halfcheetah Medium Replay distributions for the 1% setting. In these particular distributions, we can see how the application of our solution strongly strengthens the final performance. For these distributions, the TD3BC agent is able to achieve almost the same performance as the baseline, or even better for the latter case. In addition, for the Halfcheetah Medium and Hopper Medium distributions, the TD3BC algorithm again achieves almost identical performance to the baselines. In the remaining distributions, the TD3BC algorithm achieves an average of 85 percent of baseline scores.

The IQL algorithm through the expansion of the datasets, via imputed rewards, almost matches the baseline performance for the Halfcheetah Medium and Walker2D Medium Expert distributions. For the remaining environment-dataset combinations, IQL achieves around 56% of baseline results. This is a solid improvement with respect to the *1%* scenario, where IQL is just able to achieve around 15% of original scores.

In Figure 2, we show a clear example of the real effectiveness of our solution. We are comparing the D4RL scores achieved by TD3BC and IQL agents by varying the input distribution, from 1% to 5%. Moreover, we display the performance of the TD3BC and IQL agents trained on *1% + Imputed Reward* as dashed lines. On the left, we have the Walker2D Medium Replay case. For the smaller distribution (1%), the final scores are unsatisfactory for both algorithms while, on the other hand, TD3BC (blue box) is able to exploit the larger distribution (5%), thus achieving an almost-optimal score, even beating the BC (green solid line) baseline on the original dataset. However, our agent (dashed blue line) achieves a higher score. It is very important to note that to get closer to the performance of our solution, the TD3BC needs five times the number of input reward-labeled transitions. The IQL (orange box) algorithm trained on a larger distribution (5%) collapsed, thus achieving lower results than the 1% scenario. As the orange dashed line shows, our agent not only

surpasses the IQL agent trained on the larger distribution but also it almost doubles the performance of the BC (green solid line) counterpart.

On the right, we analyze the Halfcheetah Medium Expert scenario. Once again, both algorithms suffer from data scarcity, thus achieving poor performance when the 1% distribution is used. By incrementing the number of input reward-labeled transitions, the TD3BC and IQL algorithms score higher results, but not sufficiently closer to the optimal results. In particular, TD3BC (blue box) is showing better performance than the BC (green solid line) baseline, but it is way distant from our achievements (blue dashed line) which are almost optimal. IQL (orange box) trained on the broader distribution again performs worse than us (dashed orange line), but the scores obtained by both our agent and IQL are lower than agent BC (solid green line). We want to highlight that the optimal TD3BC score trained on the original dataset is 90.7 and, in this scenario, TD3BC trained on 5% of transitions from the original dataset achieves 58.45. However, our solution by leveraging *five times less reward-labeled transitions* scores 87.88.

5 Conclusion

In this paper, we have proposed a simple but effective solution for offline reinforcement learning in scenarios with extreme data sparsity in which the reward signal is incomplete. The reward model we propose is able to learn the reward signal from a very limited sample of transitions labeled with reward. Because of this, we can impute rewards for a large set of reward-free transitions, which most offline reinforcement learning algorithms are unable to exploit. The D4RL benchmark for MuJoCo tasks is used to compare our solution with TD3BC and IQL agents, where both algorithms represent the state-of-the-art for continuous control tasks. Considering an imperfect data scenario, such as having only 1% of input reward-labeled transitions, our Reward Model is able to impute the reward signal for a wider distribution of reward-free transitions. Reward signal imputation allows TD3BC and IQL agents to be trained on the entire distribution of offline experiences, which results in a substantial increase in performance, in terms of D4RL score, compared to simply applying ORL algorithms to data sparsity scenarios.

References

Martín Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *ArXiv*, abs/1701.04862, 2017.

Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, 297(C):103500, 2021. doi: 10.1016/j.artint.2021.103500.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

Nathan Gavenski, Odinaldo Rodrigues, and Michael Luck. Imitation learning: A survey of learning methods, environments and metrics. 2024.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Neural Information Processing Systems*, 2014.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Neural Information Processing Systems*, 2016.

Firas Jarboui and Vianney Perchet. Offline inverse reinforcement learning. *ArXiv*, abs/2106.05068, 2021.

Edouard Klein, Bilal Piot, Matthieu Geist, and Olivier Pietquin. A cascaded supervised learning approach to inverse reinforcement learning. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný (eds.), *Machine Learning and Knowledge Discovery in Databases*, 2013.

Ilya Kostrikov, Ofir Nachum, and Jonathan Tompson. Imitation learning via off-policy distribution matching. *ArXiv*, abs/1912.05032, 2019.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022.

Sergey Levine, Aviral Kumar, G. Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *ArXiv*, abs/2005.01643, 2020.

Dean A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In D. Touretzky (ed.), *Advances in Neural Information Processing Systems*, volume 1, 1988.

Rafael Prudencio, Marcos Maximo, and Esther Colombini. A survey on offline reinforcement learning: Taxonomy, review, and open problems. *IEEE Transactions on Neural Networks and Learning Systems*, PP:1, 03 2023. doi: 10.1109/TNNLS.2023.3250269.

Siddharth Reddy, Anca D. Dragan, and Sergey Levine. SQIL: Imitation learning via reinforcement learning with sparse rewards. *arXiv: Learning*, 2019.

Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics*, 2010.

Stuart Russell. Learning agents for uncertain environments. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, 1998.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, MA, 1998.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

Xu Wang, Sen Wang, Xingxing Liang, Dawei Zhao, Jincai Huang, Xin Xu, Bin Dai, and Qiguang Miao. Deep reinforcement learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(4):5064–5078, 2024. doi: 10.1109/TNNLS.2022.3207346.

Haoran Xu and Xianyuan Zhan. Discriminator-weighted offline imitation learning from suboptimal demonstrations. *ArXiv*, abs/2207.10050, 2022.

Maryam Zare, Parham Mohsenzadeh Kebria, Abbas Khosravi, and Saeid Nahavandi. A survey of imitation learning: Algorithms, recent developments, and challenges. *ArXiv*, abs/2309.02473, 2023.

Konrad Zolna, Alexander Novikov, Ksenia Konyushkova, Caglar Gulcehre, Ziyun Wang, Yusuf Aytar, Misha Denil, Nando de Freitas, and Scott E. Reed. Offline learning from demonstrations and unlabeled experience. *ArXiv*, abs/2011.13885, 2020.